Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Canceled)

2. (Currently Amended) The method of claim ~~1~~ 7 further comprising:

executing debug commands within the executing service.

3. (Currently Amended) The method of claim ~~1~~ 7 wherein setting the breakpoint further comprises:

locating an original instruction within the executing service to set the breakpoint;

inserting a breakpoint instruction at the breakpoint;

starting the executing service;

waiting for the breakpoint to execute;

waiting for memory fetches and configuration loads to complete; and

restoring the original instruction at the breakpoint location.

4. (Currently Amended) The method of claim ~~1~~ 7 wherein setting the breakpoint comprises:

altering an instruction within the executing service at a breakpoint location; and

invalidating a page cache of the executing service.

5. (Currently Amended) The method of claim ~~1~~ 7 wherein setting the breakpoint comprises:

setting a breakpoint register to point to a breakpoint location.

6. (Currently Amended) The method of claim ~~1~~ 7 wherein saving a minimum state comprises:

saving the executing service registers.

7. (Previously Presented) A method of debugging an executing service on a pipelined CPU architecture, the method comprising:

setting a breakpoint within an executing service;

saving a minimum state of the executing service;

setting a program counter of the executing service to point to a save stub;

setting the program counter of the executing service to point to a restore stub;

restoring the state of the executing service;

determining if registers are unstable;

if registers are unstable, saving the value of any registers that change after each pipeline cycle; and

if the breakpoint location is set on a location that uses old values of registers, saving the old values of the registers before new values are written to the registers.

8. (Original) The method of claim 7 wherein registers are scalar registers or predicate registers.

9. (Currently Amended) The method of claim ~~1~~ 7 wherein setting the program counter of the executing service to point to a save stub further comprises:

starting execution of the executing service;

executing the breakpoint;

storing configuration registers of the executing service;

saving values of scalar and predicate registers;

saving pipeline registers; and

storing a stack pointer value for a breakpoint location.

10. (Currently Amended) The method of claim ~~1~~ 7 restoring the program counter further comprising:

starting the executing service at the breakpoint.

11. (Currently Amended) The method of claim ~~1~~ 7 wherein restoring the state further comprises:

if a breakpoint location is on an instruction that does not make use of old values, restoring stable registers;

if the breakpoint location is on an instruction that does make use of old values,

restoring unstable registers, and

reloading the pipeline;

altering the program counter of the executing service to point to the breakpoint location; and

starting execution of the executing service at the breakpoint location.

12. (Currently Amended) The method of claim ~~1~~ 7 further comprising:

fetching a page of memory of the executing service into an instruction cache;

checking for a checksum error within the page of memory; and

if the executing service is set to reject the checksum error,

saving the page of memory,

inserting a breakpoint into the saved page of memory,

altering an instruction pointer to the saved page of memory, and

processing the saved page of memory.

13-18. (Canceled)

19. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to locate an original instruction within the executing service to set the breakpoint, insert a breakpoint instruction at the breakpoint, start the executing service, wait for the breakpoint to execute, wait for memory fetches and configuration loads to complete, and restore the original instruction at the breakpoint location.

20. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to alter an instruction within the executing service at a breakpoint location, and invalidate a page cache of the executing service.

21. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to set a breakpoint register to point to a breakpoint location.

22. (Currently Amended) The system of claim ~~18~~ 24 wherein the save stub is further operable to save the executing service registers.

23. (Currently Amended) The system of claim ~~18~~ 24 wherein the processing engine is further operable to flush a pipeline of a set of pipeline instructions of the executing service.

24. (Previously Presented) A system for debugging an executing service on a pipelined CPU architecture without hardware interlocks, the system comprising:

a debugger to set a breakpoint within an executing service and execute debug commands within the executing service;

wherein the debugger is further operable to set a program counter of the executing service to point to a save stub to save a minimum state of the executing service; and

a processing engine to execute the breakpoint,

wherein the debugger is further operable to set the program counter of the executing service to point to a restore stub to restore the state of the executing service, and wherein the debugger is further operable to determine if any executing service registers are unstable, save the value of any registers that change after each pipeline cycle if registers are unstable, save the old values of the registers before new values are

written to the registers, and if the breakpoint location is set on a location that uses old values of registers.

25. (Original) The method of claim 24 wherein registers are scalar registers or predicate registers.

26. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to start execution of the executing service, store configuration registers of the executing service, save values of the scalar and predicate registers, and save pipeline registers.

27. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to start the executing service at the breakpoint.

28. (Currently Amended) The system of claim ~~18~~ 24 wherein the restore stub is further operable to:

if a breakpoint location is on an instruction that does not make use of old values, restore stable registers;

if the breakpoint location is on an instruction that does make use of old values,

restore unstable registers, and

reload the pipeline;

alter the program counter of the executing service to point to the breakpoint location; and

start execution of the executing service at the breakpoint location.

29. (Original) The system of claim 28 wherein the restore stub is further operable to reload the pipeline state directly.

30. (Original) The system of claim 28 wherein the restore stub is further operable to re-execute the original instructions within the pipeline to recreate the pipeline at a time of the breakpoint.

31. (Currently Amended) The system of claim ~~18~~ 24 wherein the processing engine is further operable to:

fetch a page of memory of the executing service into an instruction cache; and

check for a checksum error within the page of memory.

32. (Currently Amended) The system of claim ~~18~~ 24 wherein the debugger is further operable to:

if the executing service is set to reject the checksum error,

save the page of memory,

insert a breakpoint into the saved page of memory,

alter an instruction pointer to the saved page of memory, and

process the saved page of memory.

33-38. (Canceled)

39. (Currently Amended) A system for debugging an executing service on a pipelined CPU architecture having a pipeline, the system comprising:

means for setting a breakpoint within an executing service;

~~means for executing service, and execute one or more no-op instructions to flush the pipeline, if there is data in the pipeline that needs to be saved;~~

~~means for executing one or more instructions for recording old values of scalar registers, if the breakpoint is set on an instruction that used the old values of the scalar registers;~~

means for saving a minimum state of the executing service;

means for setting a program counter of the executing service to point to a save stub;

means for setting the program counter of the executing service to point to a restore stub; ~~and~~

means for restoring the state of the executing service ~~using the recorded old values of scalar registers~~ ;

means for determining if registers are unstable;

if registers are unstable, means for saving the value of any registers that change after each pipeline cycle; and

if the breakpoint location is set on a location that uses old values of registers, means for saving the old values of the registers before new values are written to the registers.

40-41. (Canceled)

42. (Currently Amended) A computer readable medium comprising instructions, which when executed on a processor, perform a method for debugging an executing service on a pipelined CPU architecture having a pipeline, comprising:

setting a breakpoint within an executing service;

~~executing one or more no-op instructions to flush the pipeline, if there is data in the pipeline that needs to be saved;~~

~~executing one or more instructions for recording old values of scalar registers, if the breakpoint is set on an instruction that used the old values of the scalar registers;~~

saving a minimum state of the executing service;

setting a program counter of the executing service to point to a save stub;

setting the program counter of the executing service to pint to a restore stub; and

restoring the state of the executing service ~~using the recorded old values of scalar registers~~ ;

determining if registers are unstable;

if registers are unstable, saving the value of any registers that change after each pipeline cycle; and

if the breakpoint location is set on a location that uses old values of registers, saving the old values of the registers before new values are written to the registers.

43-44. (Canceled)

45. (Currently Amended) The method of claim ~~1~~ 7, wherein saving the minimum state further comprises saving a minimum amount of the executing service that can be restored to halt and restart execution of the service without altering the behavior of the executing service.

46-47. (Canceled)

48. (Currently Amended) The system of claim ~~18~~ 24, wherein the minimum state comprises a minimum amount of the executing service that can be restored to halt and restart execution of the service without altering the behavior of the executing service.

49-50. (Canceled)

51. (Previously Presented) The system of claim 39, wherein the means for saving a minimum state comprises means for saving a minimum amount of the executing service that can be restored to halt and restart execution of the service without altering the behavior of the executing service.

52. (Canceled)

53. (Previously Presented) The computer readable medium of claim 42, wherein saving the minimum state further comprises saving a minimum amount of the executing service that can be restored to halt and restart execution of the service without altering the behavior of the executing service.

54. (Canceled)